

# Two type-theoretical approaches to privative modification

Giuseppe Primiero<sup>1\*</sup> and Bjørn Jespersen<sup>2</sup>

<sup>1</sup> Centre for Logic and Philosophy of Science, Ghent University (Belgium)

`giuseppe.primiero@ugent.be`

<sup>2</sup> Section of Philosophy, TU Delft (The Netherlands)

`b.t.f.jespersen@tudelft.nl`

**Abstract.** In this paper we apply two kinds of procedural semantics to the problem of privative modification. We do this for three reasons. The first reason is to launch a tough test case to gauge the degree of substantial agreement between a constructivist and a realist interpretation of a procedural semantics; the second is to extend Martin-Löf's Type Theory to privative modification, which is characteristic of natural language; the third reason is to sketch a positive characterization of privation.

## 1 Introduction

The verbal agreements between constructivist/idealist and platonist/realist semantics are so numerous and so striking that it is worth exploring the extent to which there is also substantial agreement. This paper explores some of the common ground shared by the Constructive Type Theory of Per Martin-Löf<sup>3</sup> and the realist Transparent Intensional Logic of Pavel Tichý.<sup>4</sup> We focus here on the following common features:

- a notion of construction;
- a functional language;
- type theory;
- interpreted syntax.

These four features are sufficient to underpin a neutral notion of procedural semantics. Phrased in neutral terms, linguistic meaning is construed as an abstract procedure, of one or more steps, delineating what operations to apply to what operands in order to obtain a particular product as its outcome. Since the interpreted syntax is susceptible to type-theoretic restrictions, the range of admissible combinations of operations and operands is accordingly constrained. These procedures are structured constructions, each of whose constituents is an abstract object of a particular type.

---

\* Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

<sup>3</sup> See [12], [14].

<sup>4</sup> See [4], [20], [21].

In this paper we apply the procedural semantics sketched above to the problem of privative modification. We do this for three reasons. The first reason is to launch a tough test case to gauge the degree of substantial agreement; the second is to extend Martin-Löf's Type Theory to privative modification, which is characteristic of natural language; the third reason is to sketch a positive characterization of privation.

Property modification in the Montagovian tradition is a function from properties to properties. If  $M$  is a modifier and  $F$  a property, then  $(MF)$  is the property formed by applying the function  $M$  to the argument  $F$ . Thus,  $(MF)a$  is the predication of the property  $(MF)$  of the individual  $a$ . The sentential schema whose semantics we wish to study is

$$“(MF)a”.$$

The interpretation of this schema in a procedural semantics depends on the appropriate explanation of what  $M$ ,  $F$  and  $a$  are.

A full semantic theory of modification must be able to account for the following variants:

- *Subsective*:  $(M'F)a \therefore Fa$ ;
- *Intersective*:  $(M''F)a \therefore M^*a \wedge Fa$ ;
- *Modal/intensional*:  $(M'''F)a \therefore Fa \vee \neg Fa$ ;
- *Privative*:  $(M''''F)a \therefore \neg Fa$ .

The first variant is easily treated in a type-theoretical procedural semantics by standard subset formation, extending the language with quantifiers and  $\lambda$ -terms, and forming ordered pairs  $\langle M, F \rangle$  where  $F$  is the functional argument of the function  $M$  whose functional value is the modified property  $(MF)$ . The path from function and argument to value consists in deploying the operation of functional application. The second variant is less straightforward, as it requires a rule for replacing the modifier  $M$  by the property  $M^*$ .<sup>5</sup> Our conjecture is that whenever “ $Fa$ ” is an expression in a mathematical or a logical theory,  $(MF)a$  is exhausted by subsective modification, whereas for  $F$  an empirical property and  $a$  a person or an artifact, privative modification is unavoidable.

Two examples to fix ideas:

“ $a$  is a prime number”

where *prime* is a modifier of the property *number*; and

“ $b$  is a large elephant”

where *large* is a modifier of the property *elephant*. In the first example, we consider the least controversial kind of subsective modification, which goes along procedurally with subset formation: given a set of (natural) numbers, the modification of the property of being a number generates the subset of those numbers

<sup>5</sup> See [2], §4.4. The third variant will not be considered here. See [8] for discussion.

that have the additional property of being prime numbers. In the absence of obvious counterexamples, substantive modification might be conjectured to be the dominant, if not only, kind of modification needed by arithmetical discourse. Still, any semantic theory of mathematical and logical language must come with an account of modification, since the premise  $(M'F)a$  contains the modifier  $M'$ .

In empirical languages, we not only have examples like “ $b$  is a large elephant”, but also cases of privative modification, of which the following would be typical examples:

“ $b$  is a *forged* banknote”;  
 “ $b$  is *sham* jewellery”;  
 “ $b$  is a *false* friend”.

According to its definition, privation merely indicates what something is *not*, namely not an  $F$ . We do not maintain that privation is the converse operation of subsection, and it would be too strong for the constructivist to hold that privation produces the complement of the property  $F$  (because there is no such type as being  $\neg F$ ). Instead our thesis is that for the constructivist privation is an extreme case of subsection. Given a set of  $F$ s, privation will generate the null set of  $F$ 's; yet, while forming the null set of a particular property exhausts the logic of privation, its semantics is richer than that. Though both forged banknotes and railroads, say, are not banknotes, there is an intuitive sense in which forged banknotes are somehow ‘closer to’ banknotes than are railroads (or tea mugs or tax forms, etc.) The challenge is to make explicit what this (incomplete) approximation comes down to, which is to say something positive about what properties do define forged banknotes (etc.). Logically, the quest is for a definition of what it is that banknotes and forged banknotes have in common. The philosophical idea which in our view ought to inform any definition of (*forged*  $F$ ), say, is that being a forged  $F$  is as good a property as any. Hence, a procedural semantics needs to show a way of generating such a property: a constructivist semantics needs to have a way of verifying whether a particular individual has the property of being a forged banknote, and a platonist theory must be able to define the proper subset of the complement of any set of banknotes, such that the elements of that subset are forged banknotes. To do so, we characterize a privatively modified property ( $MF$ ) as having some, but not all, of the properties defining  $F$ . So there is going to be a range of forged  $F$ 's, such that those sharing more of those properties are closer approximations to  $F$ . This idea induces a sequence of properties  $G_1, \dots, G_n$  jointly defining  $F$ ; the more  $G_i$  are satisfied, the closer the approximation to  $F$ . Those forged banknotes that satisfy most  $G_i$  are virtually indistinguishable from banknotes, whereas those satisfying few are shoddy imitations (paper instead of polymer, or vice versa, wrong format, wrong colors, etc.). Still, a very poor forged banknote will nonetheless share more defining properties with a banknote than will a railroad or a tea mug.<sup>6</sup>

---

<sup>6</sup> We disregard the forger's *intention* to produce forged banknotes. We realize that by disregarding the intentions of someone designing and manufacturing technological

What is wanted, overall, is a philosophically motivated and technically workable account of privative modification interpreted within a basic neutral formulation of a procedural semantics. In particular, it must be shown what the type-theoretically constrained procedure for predicating a modified property of an individual looks like. In order to obtain such a technical result in the procedural semantics germane both to the constructivist and the realist approach to type theory, we have recourse to a procedure for subset formation. We then generate an appropriate procedure for privative modification by, accordingly, characterizing one form of subsective modification. However, Martin-Löf's and Tichý's respective theories will, in the final analysis, provide partially diverging accounts of such a procedure.

To sum up, this paper pursues two strands, one methodological, the other problem-oriented. The semantic problem is to provide a procedural account of privative modification in terms of subset formation. The methodological one concerns two different forms that a procedural semantics may take, namely the constructivism of Martin-Löf's Type Theory and the platonism of Tichý's Transparent Intensional Logic. The paper seeks to advance the research both on an ill-understood topic in semantics and the general debate of realism vs. anti-realism.

## 2 Procedural Semantics for Privative Modification

Both theories start from a notion of construction, which extends to function formation. While both operate within the confines of a typed interpreted syntax, the respective type theories work in different ways. Martin-Löf's type theory assigns a new type to each new property, laying down how to verify whether an individual has that property, whereas Tichý's type theory assigns the same type to all empirical properties of individuals. Consequently, the respective procedures for constructing a modified property are also going to differ.

### 2.1 Construction

In the constructive interpretation, predication starts by laying down all the necessary and sufficient conditions for a judgement of the form *F set* (or equiva-

---

artifacts and confining ourselves to physical properties, we are guilty of a philosophical simplification. Logically, however, a property along the lines of *being intended as a forged 100-euro banknote* can be smoothly added to the list of properties jointly defining *being a forged 100-euro banknote*. Another simplification is the absence of a priority relation over the properties jointly defining the modified one. Clearly, a realistic account of modification will discriminate between the properties that are more or less relevant to the modified property. For instance, that a forged 100-euro banknote has got the watermark right may be more relevant than getting the code number wrong. Note that in a procedural semantics like Constructive Type Theory that comes with dependent types, assumptions for hypothetical judgements are prioritized: the present formulation is therefore a simplification where presuppositions and assumptions are all introduced at the same level of relevance.

lently  $F \text{ prop}$ ) to be formulated: such a type declaration is justified in terms of a judgement  $f : F$  that shows a constructor for that set, and an equality judgement  $f = f' : F$ , to ensure canonicity for that element. From categorical judgements of the form  $f : F$ , one extends the language to hypothetical judgements as expressions of the form  $F' \text{ type}[x : F]$  which can be intended as a relation between types, corresponding to *functional abstraction*. The justification of such a form of judgement is given by saying that  $F'$  is a type whenever an appropriate substitution is performed by a certain canonical constructor  $f$  in the type  $F$ . The functional extension of the language is crucial to expressing implicational and quantified formulae. The obtained type is that of functions from type  $F$  to type  $F'$  taken as objects. If  $F$  is a type, the construction of a new type is possible by considering  $F'$  a family of sets over some  $x : F$ , such that  $F'[x : F]$  is also a type. A function can, therefore, be construed as the judgement regarding a certain object  $F' \text{ type}$  based on the prior judgements that any  $F$  is a type and that any  $f$  is an object or construction for that type ( $f : F$ ).<sup>7</sup> This standard formulation is easily generalized to more than one assumption. For each type computational rules for formation, introduction, elimination and equality are defined. The theoretical starting point of Martin-Löf's type theory is, therefore, the justification of a typed expression in terms of its *proof-object* and the reduction of truth-conditions to assertion-conditions.

Similarly, the procedural aspect of Tichý's theory is given by the fact that the  $\lambda$ -terms of application and abstraction do not denote, respectively, the *result* of applying a function to an argument or arranging two sets of entities as functional arguments and their values. Rather, in TIL, they denote, respectively, the very *procedure* of applying a function to an argument and of forming a function. Application is called *Composition* in TIL and is encoded thus:  $[X_0 X_1 \dots X_n]$ , where  $X_0$  is a construction of a function,  $X_1, \dots, X_n$  constructions of its arguments and  $[ \ ]$  the procedure of functional application. Abstraction is called *Closure* in TIL and is encoded thus:  $[\lambda x_1 \dots x_n Y]$ , where  $x_1, \dots, x_n$  construct arguments,  $Y$  constructs values of a function and  $[\lambda x_1 \dots x_n Y]$  is the procedure of functional abstraction.<sup>8</sup>

## 2.2 Functional Language

Constructively, a property  $F$  is introduced as a basic type in a valid predication by presenting some individual that instantiates it ( $f : F$ ). A function defined over  $F$  is a type. Its construction corresponds to a *propositional function*, which is here used to give a formal treatment of property modification. The neutral formulation  $(MF)a$  of an individual  $a$  instantiating the modified property  $(MF)$

<sup>7</sup> See [16], §1.7.

<sup>8</sup> Two other constructions are *Trivialization* and *Variable*. Trivializations can be dispensed with here, since we do not need to *mention* constructions; we only *use* them to obtain the entities they construct. For now, think of variables as one-step procedures for obtaining an entity relative to a sequence of assignments of entities to variables. See [2], §§1.1-1.3.2, §2.6.1.

is constructively expressed as a function  $M$  such that for every element in the set  $F$  taken as argument, it returns a function  $M(x)$ , formally  $M(x)[x:F]$ . To preserve the functional aspect of  $M$  in the constructive notation, we will refer to  $M(F)$  *type* as the modified type satisfied by some  $f:F$ ; correspondingly,  $(MF)a$  will be expressed by  $M(f)$ . Standard modification of a property  $M(F)$  is given, therefore, by functional abstraction and it produces subset formation  $\{x:F \mid M(x)\}$ . The case of privative modification is no exception to this general interpretation: a privative modifier will still take as arguments elements in a basic type  $F$ , hence occurring at the level of extensions. It differs from a standard functional type (and standard subset formation) in that it does not define a set of individuals of the basic type, because its arguments no longer instantiate the original property  $F$ . Rather, the range of this modifier will consist of functions from the basic type  $F$  to the empty set. This shows that constructive privation represents a special case of standard subsection, specified by requiring extra conditions. That the range of the privative modifier is a set of functions of the appropriate type – rather than individuals – can be seen as introducing a type of higher order. The constructivist sense of the predicate “is a forged banknote” is, therefore, the procedure of applying to the set of individuals that are typed as banknotes the modifier *forged* as a propositional function, whose range corresponds to functions pointing at another set of individuals (an empty one). The bottom-up approach characteristic of the constructive philosophy is preserved, so that the Introduction Rule uses a construction  $f:F$  as a premise to define a privatively modified  $F$  in terms of the empty set of  $F$ ’s.

The functional language of TIL is cast within a ramified type hierarchy encompassing a simple type theory, relative to which each entity of the ontology of TIL receives a type. The entities are organized into a bi-dimensional typed universe. One dimension is made up of non-constructions, the other of constructions. On the ground level of the type hierarchy there are non-constructional entities unstructured from the procedural point of view belonging to a *type of order 1*. Given a so-called *epistemic* (or, equivalently, *objectual*) *base* of *atomic types* ( $\alpha$ -truth values,  $\iota$ -individuals,  $\tau$ -reals doubling as times,  $\omega$ -possible worlds), the induction rule for forming *functional types* is applied: where  $\alpha, \beta_1, \dots, \beta_n$  are types of order 1, the set of partial mappings from  $\beta_1 \times \dots \times \beta_n$  to  $\alpha$ , denoted ‘ $(\alpha\beta_1 \dots \beta_n)$ ’, is a type of order 1 as well. Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted ‘ $*_1$ ’. The type  $*_1$  together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, of type  $(\alpha\beta_1 \dots \beta_n)$ , involving  $*_1$  in their domain or range, is a *type of order 2*. Constructions belonging to a type  $*_2$  that construct entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*; and so on *ad infinitum*.<sup>9</sup>

Tichý’s theory of modification proceeds, therefore, in a strictly top-down manner. First, a modified property is constructed according to the procedure of functionally applying a modifier  $M$  to a property  $F$ , and only then is the modified property  $(MF)$  predicated of an individual  $a$ . What gets predicated

<sup>9</sup> See [2], §1.3.2.

of an individual is, strictly speaking, an *extensionalized* property, which is a function from individuals to truth-values.

An intensional entity is any function (mapping) whose domain is in the logical space of possible worlds. For most purposes, TIL takes an intension to be a function from logical space to a function from times to entities, in the manner well-known from possible-world semantics enriched with temporal parameters. Thus, an empirical property of individuals is a function from logical space to a function from times to sets of individuals, where a set of individuals is a characteristic function from individuals to truth-values. Hence, given a particular world/time pair  $\langle w, t \rangle$ , it is either true or false that a given individual  $a$  is a member of the set that is the extension of the property at  $\langle w, t \rangle$ . Formally, the type of a property is  $((\omega)\tau)\omega$ , abbreviated ‘ $(\omega)\tau\omega$ ’. The TIL abbreviation of a modified empirical property being predicated of an individual will be of the form  $\lambda w \lambda t [[MF]_{\tau\omega} a]$ .

The two theories will formalize “No forged banknote is a banknote” thus:

**CTT:**  $\lambda x. (forged(x)) : El(banknote(x)[banknote : El(\{\}) set; x : El(\{\})])$   
**TIL:**  $\forall w \forall t [\forall x [[[forged\ banknote]_{wt} x] \rightarrow \neg [banknote_{wt} x]]]$

### 2.3 Interpreted Syntax

The procedural way of generating privatively modified properties is based on the fact that the type-theoretical syntax is interpreted.

Constructive Type Theory can be seen as one of several foundational systems for predicative constructive mathematics,<sup>10</sup> but its additional value is represented by a meaning theory which extends and refines the Brouwer-Heyting-Kolmogorov interpretation of intuitionistic logic.<sup>11</sup> Besides providing a constructive foundation for mathematics,<sup>12</sup> CTT is also used as a proper theory of reasoning and knowledge, an interpreted system whose objects are equipped with meanings.<sup>13</sup> This is obtained by implementing the Curry-Howard isomorphism, by which types are intended as polymorphic categories of predication, carrying an internal meaning that can be explicitated in terms of propositions (for which proofs are the appropriate constructors) or sets (correspondingly constructed by their elements). This property can be adapted to the interpretation of natural language semantics,<sup>14</sup> where reference is generally construed as the relationship between nouns or pronouns and the objects that are named by them. In the constructive procedural semantics every object comes embedded within its meaning category, in a relation (that does not generate a vicious circle) according to which a type gains its meaning (is justified) from its constructor, and the constructor is meaningfully expressed whenever accompanied by its type (“no

<sup>10</sup> Constructive set theory, explicit mathematics and predicative topos are other examples of systems of constructive mathematics.

<sup>11</sup> Cf. [11], [12], [16], ch.1.

<sup>12</sup> Cf. [12], p. 1, §1.

<sup>13</sup> Cf. [16], ch.1.

<sup>14</sup> See [18].

entity without a type”).<sup>15</sup> The input of the characteristic function for any set is positively given by appropriate constructors which can always be checked; any propositional function operates on the corresponding predications that work as its arguments. As a result, any expression occurring in one of the computational rules comes embedded with types that yield meanings, and each meaning category is reduced to the corresponding syntactical construction procedure.

The syntax of TIL (its formal ‘language of constructions’ in which constructions are encoded) is inherently interpreted because both constructions and the entities they construct cannot be introduced without typing them first.<sup>16</sup> A *semantic analysis* of a piece of language executed in accordance with TIL proceeds along the following three steps.<sup>17</sup> First, *type-theoretic and logical analysis*: all and only logical entities (operations and their operands) being denoted by sub-expressions occurring in the overall expression under analysis receive a type, which may be drawn from the simple or ramified type hierarchy. Second, *synthesis*: the constructions of the entities mentioned are executed in accordance with the logical operations made explicit by the logical analysis in order to unveil the entity denoted by the overall expression. Third, *type checking*: by means of an annotated tree it is checked whether the type assignments check out.<sup>18</sup>

### 3 Constructive Privative Modification

Standard subsets are used in the type-theoretical setting in order to express a type that is defined by comprehension in the range of another type. Constructively, this corresponds to nothing other than a propositional function from one type to another: its construction requires the definition of the argument type in terms of necessary and sufficient conditions for a canonical element and an equality function defined on it. An object of the function type is obtained by an abstraction rule abstracting a variable from an expression, obtaining a function  $(x):F'(x:F)$  which requires  $f'$  object of the type  $F'$ , depending on a variable  $x$  ranging over the type  $F$ , so that one abstracts  $f'$  with respect to  $x$ . This rule of functional abstraction is equivalent to Church’s  $\lambda$ -abstraction. To know that the preceding rule is correct, it must be shown that when this function is applied to any object of the type  $F$ , one gets an object of the type  $F'(f/x)$ , a typed version of  $\beta$ -conversion.<sup>19</sup> The range of the function type will be obtained by functional application to the elements of the argument type. In this way one obtains the subset of elements in  $F$  satisfying  $M$ :

$$\frac{F \text{ set} \quad M(x)[x:F]}{\{x:F \mid M(x)\}} \quad \text{Standard Subset Formation Rule}$$

<sup>15</sup> See also [15].

<sup>16</sup> See [2], §1.5.1, §2.1.2.

<sup>17</sup> See [2], §2.1.1.

<sup>18</sup> See [10] for details.

<sup>19</sup> See [16], §1.8. For an analysis of functions and types and the reference of abstract terms, see [17].



By the side condition on canonical elements, if  $f = f'$  and  $M(x)$  is true for some  $x : F$ , one obtains equal canonical constructions of the set  $\{x : F \mid M(x)\}$  when  $f$  or  $f'$  are used as input of  $M$ . That is, since every propositional function is extensional in the sense that it yields equal types when applied to equal elements, it follows from  $f = f' : F$  and  $M(x) \text{ type}[x : F]$  that  $M(f)$  and  $M(f')$  are equal types. Consequently, from the requirement that  $M(f)$  be true, we immediately get that also  $M(f')$  is true. It is clear from the formulation of the function formation rule and its conversion that the language in use is extended to quantifiers, so that the function ranges over all elements of a given family of sets.

The use of subset formation for an arbitrary property  $F$  (e.g. *banknote*) and a *privative* modifier  $M$  (e.g. *forged*) is not entirely correct, however. To preserve the constructive interpretation also for the case of privative modification, it is required that the meaning of  $M(F) \text{ type}$  be given by some (canonical)  $M(f)$ , constructed by using a canonical  $f : F$ . By using standard subset formation, the modifier type  $M$  will yield a subset of the set of *canonical*  $F$ 's. Since a *privative* modifier  $M$  is intended as a modification procedure that changes entirely the range of its input, an alteration is needed. Because a forged banknote is not a banknote in the first place, the privative modifier *forged* cannot be interpreted as a propositional function from the canonical set of banknotes to one of its (canonical) subsets. For this reason, one needs to define privative modification as an extreme version of subsection. The obvious intuition is that the basic argument  $F \text{ set}$  needs to be modified whenever used as an input of the privative modifier  $M$  in a way that allows us to turn every  $x : F$  into an element of the function from  $F$  to the empty set. The first step towards obtaining such a procedure is to define appropriate constructions of the empty set and of the function from a set to the empty one, returning the empty set of elements in that set. The empty set is introduced by declaring the following constants:<sup>20</sup>

$$\begin{aligned} \{\} &: \text{Set}; \\ \text{case}_{\{\}} &: \text{El}(Z(x)) \ [(Z : \text{El}(\{\})) \text{Set}, x : \text{El}(\{\})]. \end{aligned}$$

The first constant simply declares the collection with no elements to be a set; the case step gives the empty set of  $Z$ 's elements, by applying a set  $Z$  to any element  $x$  on condition that  $Z$  be an element of the collection of empty sets, and  $x$  an element of any set in that collection. Both of these constructions are crucial to the formulation of the privative modifier  $M$ : such a modifier uses a canonical set construction for  $F \text{ set}$  and it returns an element of the empty set of that specific set  $F$ . The idea is therefore that, if  $F$  is a set, then the application of the modifier  $M$  to any  $x : F$  yields the set of  $M(x)$  such that holds under conditions that  $x : \text{El}(\{\})$  and  $F : \text{El}(\{\}) \text{ set}$ :

Privative Subset Formation Rule

$$\frac{F \text{ set} \quad M(x)[F : \text{El}(\{\}) \text{ set}; x : \text{El}(\{\}); \text{El}(F(x))]}{\{x : F \mid M(x)\}}$$

<sup>20</sup> Cf. [14], p.151.

This construction defines a function  $M$  over the set  $F$ , which does not give a subset of its canonical elements: given  $x:F$  as input of this function,  $M(x)$  returns the empty set of  $f$ 's as its output, instead of either an arbitrary empty set or a (constructively inadmissible) canonical element for  $\neg F$ . There is constructively no way to give a definitional procedure for a negative type such as *the set of non-banknotes*, because its conditions cannot be canonically specified, in case *the set of non-banknotes* should include everything that does *not* satisfy the conditions for being a banknote. But there is nothing wrong with giving the assertion conditions for a function that takes any element in the set of banknotes to the complement of such a set, because in this case it is completely specified what the conditions for its input are, and the function only requires that those conditions remain (entirely or partially) unsatisfied.

When the Privative Subset Formation Rule is applied to the example of *forged banknote*, one starts from the set of banknotes and then, by applying the appropriate conditions on *banknote set*, one wishes to obtain the empty set of banknotes:

$$\frac{\text{banknote set} \quad \text{forged}(x)[\text{banknote}:El(\{\}) \text{ set}; x:El(\{\}); El(\text{banknote}(x))]}{\{x:\text{banknote} \mid \text{forged}(x)\}}$$

It is essential, therefore, to operate with *typed* empty sets.

One basic problem for privative modification treated as output of the empty-set function is laying down the distinction between the output of  $M(F)$  – for  $M$  some privative modifier like *forged* and  $F$  an argument, e.g. *banknote* – and any other empty set: what is the difference between constructing the empty set of banknotes in terms of the set of forged banknotes and any other way of constructing a set none of whose elements is a banknote? This problem is constructively solved by putting forward an appropriate equality rule governing  $M(F)$  with respect to the set  $F$ :

Equality Rule on Sets

$$\frac{F \text{ set} \quad F = F' \text{ set} \quad M(x)[F = F':El(\{\}) \text{ set}; x:El(\{\}); El(F = F'(x))]}{\{x:F = F' \mid M(x)\}}$$

By this rule for any equivalent set taken as argument of the modifier, the same empty set is obtained. This also allows establishing that for any set  $G$  with its own constructor  $g \neq f:F$  the modifier  $M(x)[x:G]$  shall return a different empty set (namely, the empty set of  $G$ 's, different from the empty set of  $F$ 's). This obviously allows defining the difference between  $M(F)$  (*forged banknotes*) and  $G$  (*railroads*, say) as empty sets of banknotes in a different sense: the former will, strictly speaking, be the set of function constructors from the set of banknotes to the empty set; the second set will contain no constructor of the set of banknotes at all, hence being empty with respect to any such individual.

The introduction rule is meant to instantiate the procedure which, starting from a typed object, returns a privatively modified one:

Introduction Rule

$$\frac{f:F \quad m:M(f)[F:El(\{\}) \text{ set}; f:El(\{\}); El(F(f))]}{f:\{x:F \mid M(x)\}}$$

where  $F$  can be taken to be the set of banknotes and  $f$  an instance of that set, and  $M$  the modifier *forged*.

Equality on Introduction Rule:

$$\frac{f=f':F \quad m:M(f)[F:El(\{\}) \text{ set}; f:El(\{\}); El(F(f))]}{f=f':\{x:F \mid M(x)\}}$$

In the introduction rule one starts from the premise that a canonical element  $f$  in the set  $F$  is given; provided  $M(f)$  is true, i.e. there is a canonical element  $m$  of the set of functions from  $F$  to the empty set, we know that  $f$  will yield a canonical element in the set of modified  $F$ 's when taken as the argument of the empty-set function of  $M(F)$ . By the associated equality rule, if  $f = f'$  are elements in  $F$ , and if there is an  $m$  such that  $M(f)$  is true,  $f$  and  $f'$  will yield canonical elements in the set of modified  $F$ 's; and from  $f = f':F$  and  $m:M(f)$  it follows that  $m:M(f')$ . Notice that according to the constructive requirement on the introduction rule, in order to form the set of modified  $F$ 's, one needs to know at least one instance  $m:M(f)$ , and because the latter relies on a function applied to  $f$ , it is a further presupposition that  $f$  be known. For example, in the case of *forged banknote*, in order to display or recognise a forged banknote one needs to be able to lay down the conditions for knowing what a banknote is.

The set of rules is rounded off by an appropriate elimination rule, which makes one able to specify how to extract a modified property from its corresponding set. Notoriously, formulating an elimination rule for the subset theory is a difficult matter. It is impossible to give in constructive type theory an elimination rule that captures the way one has introduced elements in a subset, because there is no explicit construction of the element  $m:M(f)$  for a standard subset  $\{x:F \mid M(x)\}$ .<sup>21</sup> In the case of privative modification, the elimination rule is supposed to formalise the procedure which, starting from an element of a privatively modified property (*forged banknote*, say), will return another modified element defined over the former; this means that variables will occur bound in the second construction. The informal meaning of the elimination rule is to enable positive predication for privatively modified entities. Saying that a banknote can be identified by ascertaining that it reacts to ultra-violet lamps emitting light at around 365 nanometres<sup>22</sup> can be rephrased by saying that a forged bank-

<sup>21</sup> Cf. [19] for a full explanation, the solution proposed and the consequences for the deductive power of the theory.

<sup>22</sup> Pamphlet of the Bank of England, downloadble at [http://www.bankofengland.co.uk/banknotes/kyb\\_lo\\_res.pdf](http://www.bankofengland.co.uk/banknotes/kyb_lo_res.pdf).

note will fail to react to uv-lamps emitting light at around 365 nanometres; similarly, one may want to state of a false friend that he or she is a seasoned liar, or that sham jewellery is an “abomination [...], a lie, a pretension”.<sup>23</sup> In the following, let  $\Delta$  abbreviate the condition on a privatively modified set as given by the premise in its introduction rule. In the corresponding elimination rule, one starts from an instance  $f$  of a privatively modified property  $M(F)$ . Its extension is constructed under the assumption that  $x : \{f : F \mid M(f)[\Delta]\}$ , so that another function  $f'$  of type  $M'$  is constructed which takes  $x$  as argument and forms the new type  $M'(x)$ : by substituting  $f$  in the free occurrences of  $x$  in  $M'(x)$ , one concludes that  $f'(f)$  is an element of the newly modified type  $M'(f)$ :

Elimination Rule

$$\frac{f : \{x : F \mid M(x)[\Delta]\} \quad f'(x) : M'(x)[x : F, m : M(x)]}{f'(f) : M'(f)}$$

### 3.1 Degrees of Modification

Standard typing rules do not as yet say anything relevant about the sense in which modification comes in degrees, given that there are different sorts of forged banknotes. For example, in the light of a description of a banknote as a green piece of polymer with an hologram printed on it, there are different ways in which a forged banknote may be forged: it may be a piece of polymer which is either not green or lacks the appropriate hologram, it can be a green piece of something other than polymer with or without an hologram printed on it, and it can lack two of the three defining properties. All in all, an individual that lacks all three properties fails to qualify as a forged banknote. We shall explain these differences by introducing a formal notion of degrees of modification.

The use of *dependent* types has been shown to be crucial to the definition of the subset formation rule, both in its standard format and its privative variant. We want now to make a dependency relation explicit also for the argument of the modifier function, which will make it possible to differentiate among privatively modified  $F$ 's. Take

$$F \text{ set}[x_1 : F_1, \dots, x_n : F_n]$$

to be the formal way of saying that  $F$  is a canonical set whenever each  $x_i : F_i$  is a type-theoretical expression satisfied by an appropriate element  $x_i/f_i$ , where each  $F_i$  is a definitional property of  $F$ .<sup>24</sup> The rule for defining the privative modifier

<sup>23</sup> From the *Routledge Manual of Etiquette*, 2007, p. 175.

<sup>24</sup> In the present treatment of type-theoretical predications, we are referring to standard types requiring a finitistic formulation of a dependency relation from a context of assumptions. In [13], a non-standard extension of intuitionistic type theory with infinite objects was introduced, which represents a generalization of the finitistic frame, relying on the latter for justification. In [1] this non-standard extension of the theory is used to explore the formalization of contexts for universal, existential

can be analytically formulated with respect to its application to the definitional properties  $F_i$  of  $F$ :

Dependent Privative Subset Formation

$$\frac{F \text{ set}[x_1:F_1, \dots, x_n:F_n] \quad M(x)[F_i:El(\{\}) \text{ set}; x:El(\{\}); El(F_i(x))]}{\{x:F \mid M(x)\}}$$

where  $1 \leq i \leq n$ . This new rule says that  $M(x)$  is a modified  $F$  in view of the empty set of  $F_i$ , for every  $\bigvee F_i \in F$  up to  $\bigwedge F_i$  defining  $F$ , that is by privation with respect to some – up to all bar one – of its definitional properties.

Depending on the selection and combination of  $F_i$ , one obtains different *degrees of modification*. A standard recursive definition of the factorial of the integer  $n$

$$n! = \begin{cases} 1, & \text{if } n = 0 \\ n(n-1), & \text{if } n \geq 1 \end{cases}$$

is used in the following for the standard combinatorial result of  $d$  elements extracted from  $n$ :

$$C_n^d = \frac{n!}{d! \cdot (n-d)!}$$

In the following we shall use  $n$  to indicate the number of  $F_i$  occurring in the dependency context of definitional properties of  $F$ , so that we shall call the degree  $d$  of modification  $M$  of a property  $F$  the number of  $n$  definitional properties of  $F$  with respect to which a privative modifier is applied.

By the combinatorial result given above for  $C_n^d$ , the following can be easily stated:

- there will be  $n$  distinct modifications of degree  $d = 1$ , corresponding to the privation of  $x:F$  with respect to  $F_i$  for some  $i \in n$  in the set of conditions for  $F \text{ set}$ ;
- there will be a combinatorial number of distinct modifications of degree  $d = i < n$  in view of the rule for  $C_n^i$ , corresponding to the privation of  $x:F$  with respect to the union  $\bigcup\{F_1, \dots, F_i\}$ ,  $2 \leq i < n-1$  in the set of conditions for  $F \text{ set}$ .

Following this rule, an individual determined by 10 properties will accommodate a total of 198,720 possible combinations of modification, counting all the

---

and intensional predication. Moreover, it is also used to model the negation of predicates at one stage (or more) in the infinite dependent structure of contexts, which recalls the notion of unsatisfied conditions introduced here to formalize the notion of degree of modification. As mentioned in the Introduction, we are relying on the simplification that elements in the dependency context come without any priority relation.

modifications of one property, those of two properties and so on, up to counting 10 possible combinations of modification involving 9 properties (obtained by the calculation  $3,628,800/362,880 = 10$ ). For a simple example, consider the definitional presentation of the set of banknotes introduced above, for which three different modifications of degree 1 are possible, making forged banknotes forged due to their being deprived of just one defining property:

$$\frac{\text{banknote set}[\text{polymer}, \text{green}, \text{hologram}] \quad \text{forged}(x)[F_i : \text{El}(\{\}) \text{ set}; x : \text{El}(\{\}); \text{El}(F_i(x))]}{\{x : \text{banknote} \mid \text{forged}(x)\}}$$

where  $F_i$  is a variable for any of the properties of being made of polymer, of being green or of presenting an hologram. A modification of degree 2 would take into account two defining properties; as a result, an instance of the following constructor would be a forged banknote by failing to be made of green polymer (or any other combination):

$$\frac{\text{banknote set}[\text{polymer}, \text{green}, \text{hologram}] \quad \text{forged}(x)[F_{i,j} : \text{El}(\{\}) \text{ set}; x : \text{El}(\{\}); \text{El}(F_{i,j}(x))]}{\{x : \text{banknote} \mid \text{forged}(x)\}}$$

where again  $F_{i,j}$  instantiate two defining properties.

### 3.2 Iteration of Modifiers

The formulation of degrees of modification enables us to make comparisons among different instances of the same modified type. In particular, it enables us to express, in the metatheory, that a particular modified set is at a certain degree of approximation to its original counterpart. In the case of *forged banknote*, privative modification of degree 1 will be a closer approximation to *banknote* than a privative modification of degree 2. This squares with natural-language predicates like ‘is a well-made forged banknote’, whose use presupposes various degrees to which a forged banknote may succeed in passing for what it is a forgery of.

This remark leads directly to the next case we want to analyse, namely the iteration of modifiers. The modifier *well-made* needs to qualify *forged banknote*, otherwise one ends up with *((well-made forged) banknote)*: as we already know, a forged banknote is not a banknote, hence *well-made* is subsective with respect to a set formed by applying a privative modifier to a non-modified set.<sup>25</sup> For the iteration to be such that, given a set of forged banknotes, one extracts only the well-made ones, one has to be sure that the construction of *(well-made (forged banknote))* uses a correct application of different subset formation rules.

<sup>25</sup> Brackets are used as scope indicators. Note that if *well-made* is to modify *forged*, then because the latter is a first-order modifier (modifying, as it does, a non-modifier), the former must be a higher-order modifier like, e.g., *very*. See [7] for discussion of higher-order modification.

Consider the by now well-known construction of *forged banknote* and let us abbreviate again the additional conditions on the privative subsection as  $\Delta$ . Now the construction of (*well-made (forged banknote)*) is of the following form:

$$\frac{\frac{\text{banknote set} \quad \text{forged}(x)[\Delta]}{\{x:\text{banknote} \mid \text{forged}(x)\}} \quad \text{well} - \text{made}(x)[x:\text{banknote} \mid \text{forged}(x)]}{\{x:\text{banknote} \mid \text{well} - \text{made} \times \text{forged}(x)\}}$$

This construction first applies the privative subset formation rule and then applies the standard subset formation rule to the resulting set of functions, thus obtaining the cartesian product of two families of functions over correctly defined sets.

On the other hand, the construction of (*well-made forged banknote*) is an illegitimate one. The predicate ‘is a (well-made forged) banknote’ does not split the application of the modifiers into two steps, hence the formal construction combines by cartesian product the standard subsecutive modifier and the privative subsection. Because the subsecutive modifier *well-made* has as its argument the categorical set *banknote* (it can be seen as the set of identity functions from each element in that self to itself), whereas the argument of the privative modifier *forged* is constructed on condition of being arguments for functions defined over an empty set, the resulting construction is ill-defined:

$$\frac{\text{banknote set} \quad \text{well} - \text{made}(x)[x:\text{banknote}] \times \text{forged}(x)[\Delta]}{\{x:\text{banknote} \mid \text{well} - \text{made}(x) \wedge \text{forged}(x)[\Delta]\}}$$

A specific case of iteration of modifiers is represented by iteration of privative modifiers. This kind of iteration avoids the problem of the previous case, because in both cases the modifiers are privative, hence their arguments are both constructed under appropriate conditions. The iteration will give the cartesian product of the sets of functions that are arguments of the modifier. The following construction is an example of a formation rule regulating *burned forged banknote*:

$$\frac{\frac{\text{banknote set} \quad \text{forged}(x)[\Delta]}{\{x:\text{banknote} \mid \text{forged}(x)\}} \quad \text{burned}(x)[\text{forged}(x)[\Delta]]}{\{x:\text{banknote} \mid \text{forged} \times \text{burned}(x)[\Delta]\}}$$

*Burned* is privative because a burned  $F$  is not an  $F$ , though it originally was an  $F$ . Two privative modifiers do not cancel each other out, such that a burned forged banknote would be a banknote.<sup>26</sup> Furthermore, though both *forged* and *burned* are privative, their logical behaviour does not overlap entirely. In particular, “ $a$  is a burned banknote” is an example of resultative predication<sup>27</sup> while “ $a$  is a forged banknote” is not. From  $a$  being a burned banknote it follows that

<sup>26</sup> See [8] for further discussion.

<sup>27</sup> See [3], p. 226ff.

$a$  is not a banknote (because a pile of ashes does not make a banknote), but it is presupposed that  $a$  started out as a banknote (otherwise there would have been no banknote to burn). So *burned* comes with a dynamic dimension that *forged* lacks: a forged banknote was never a banknote and only remains an approximation to one. (As for  $a$  being a well-made forged banknote, the degree to which  $a$  qualifies as being well-made is a reflection of the quality of the craftsmanship of the forgery.)

## 4 Realist Privative Modification

### 4.1 Predication of modified properties

A property is an intensional entity of type  $(((\iota\iota)\tau)\omega)$ , abbreviated ‘ $(\iota\iota)_{\tau\omega}$ ’, which is a function from worlds ( $\omega$ ) to functions from times ( $\tau$ ) to sets of individuals  $((\iota\iota))$ . A property modifier, by contrast, is an extensional entity, because it is not indexed to possible worlds. Instead it is a function-in-extension between two intensions. Since a property modifier is a function that takes one property to another, its type is  $((\iota\iota)_{\tau\omega}(\iota\iota)_{\tau\omega})$ . So in order to construct a modified property, the procedure of functional application (Composition) is called for:

$$[\textit{modifier property}]$$

The predication of a property of an individual goes via two instances of functional application. First, the relevant property is extensionalized so as to obtain a set from a property. Second, the set is applied to the individual to obtain a truth-value. The philosophical motivation is that individuals exemplify empirical properties only relative to worlds and times.<sup>28</sup> Schematically, predication is this Closure:

$$\lambda w \lambda t [\textit{property}_{wt} a]$$

This Closure, which constructs a possible-world proposition (a function from worlds to functions from times to truth-values), would be the logical form of the sense of a sentence like, “ $a$  is a banknote”.

The schema of the predication of a modified property of  $a$  is this Closure:

$$\lambda w \lambda t [[\textit{modifier property}]_{wt} a]$$

This Closure would be the logical form of the sense of a sentence like, “ $a$  is a forged banknote” or “ $a$  is a burned banknote”.

If the property constructed by  $[\textit{modifier property}]$  is itself modified, the resulting predication looks like this:

$$\lambda w \lambda t [[\textit{modifier}' [\textit{modifier property}]]_{wt} a]$$

This would be the form of, say, “ $a$  is a burned forged banknote” or “ $a$  is a well-made forged banknote”. In all three cases the semantic analysis culminates in the assignment of a propositional construction to a sentence as its sense.

<sup>28</sup> See [6] for details.



## 4.2 The requisites of privation

True to its top-down approach, TIL accounts for a property like *being a forged banknote* in terms of other properties being ‘stacked upon it’, to wit, the set of properties that are individually necessary and jointly sufficient for an individual to have that property. Such a set is called the *essence* of the property in question, and each element is called a *requisite*.<sup>29</sup> The type of a requisite, when a relation-in-extension between two properties, is  $(o(o\iota)_{\tau\omega})(o\iota)_{\tau\omega}$ , while the type of the essence of a property is  $((o(o\iota)_{\tau\omega})(o\iota)_{\tau\omega})$ : the *essence* function takes a property to the set of properties that are its requisites. Formally,  $F$  being of type  $(o\iota)_{\tau\omega}$  and  $p$  ranging over the same type, these two constructions converge in the same set of properties:

$$[essence\ F] = \lambda p\ [Req\ p\ F]$$

The requisite relation is defined in the following manner. Let  $X, Y$  be intensional constructions such that  $X, Y$  are first-order constructions ranging over the type  $(o\iota)_{\tau\omega}$  (i.e.  $X, Y$  are property variables) and let  $x$  range over  $\iota$ .<sup>30</sup> Then:

$$[Req\ YX] = \forall w\forall t\ [\forall x\ [[True_{wt}\lambda w\lambda t\ [X_{wt}x]] \rightarrow [True_{wt}\lambda w\lambda t\ [Y_{wt}x]]]]$$

Gloss *definiendum* as, “ $Y$  is a requisite of  $X$ ”, and *definiens* as, “Necessarily, at every  $\langle w, t \rangle$ , whatever  $x$  instantiates  $X$  at  $\langle w, t \rangle$  also instantiates  $Y$  at  $\langle w, t \rangle$ .”

Logically, privation comes down to, say, *being a banknote* and *being a forged banknote* having an empty intersection at every  $\langle w, t \rangle$ . This is obtained thus:

$$[Req\ \lambda w\lambda t\ \neg[banknote_{wt}\ x]][forged\ banknote]$$

We say that the property constructed by  $[forged\ banknote]$  has, *inter alia*, the requisite property constructed by  $\lambda w\lambda t\ \neg[banknote_{wt}\ x]$ . This is to say that if, at some  $\langle w, t \rangle$  or other, an individual  $x$  is in the extension of  $[forged\ banknote]$  then  $x$  is in the extension of  $\lambda w\lambda t\ \neg[banknote_{wt}\ x]$ .

Hence, the proposition that *not being a banknote* is a requisite of *being a forged banknote* is equivalent to the proposition constructed thus:<sup>31</sup>

$$\forall w\forall t\ [\forall x\ [[forged\ banknote]_{wt}\ x] \rightarrow [\neg[banknote_{wt}\ x]]]$$

What is special about the sort of non-banknote that is not a tea mug, a railroad or a tax form, but a forged banknote? Given a  $\langle w, t \rangle$ , the set constructed by  $[banknote_{wt}]$  will have a complement in which we find tea mugs and all the rest, including forged banknotes, but the set constructed by  $[forged\ banknote]_{wt}$

<sup>29</sup> See [2], §4.4. Requisites play pretty much the same role as do *presuppositions* in constructivism.

<sup>30</sup> See [2] §4.1, def. 4.1. See also §4.1 for *True*, which is the propositional property of being true at  $\langle w, t \rangle$ .

<sup>31</sup> For the record, ‘ $\forall y$ ’ abbreviates ‘ $[^0\forall[\lambda y]]$ ’,  $y$  ranging over an arbitrary type  $\alpha$ ,  $\forall$  a function of type  $(o(o\alpha))$ , and  $^0\forall$  being the Trivialization of this function.

will be a well-defined proper subset of that complement. To define the notion of the subset of forged banknotes within the set of non-banknotes, we need to express that no forged banknote is a banknote and that some non-banknotes are forged banknotes:

$$\forall w \forall t \ [ [ [All \ [forged \ banknote]_{wt}] [\lambda x \neg [banknote_{wt} \ x]]] \wedge \\ [ [Some \ [\lambda x \neg [banknote_{wt} \ x]]] \lambda x \neg [ [forged \ banknote]_{wt} \ x]] ] ]$$

We invoke the quantifiers *All*, *Some*, here of type  $((o(o\iota))(o\iota))$ .<sup>32</sup> *All* is the function from the set constructed by  $[F_{wt}]$  to the set of all those sets that contain the set constructed by  $[F_{wt}]$  as a subset. *Some* is the function from the set constructed by  $[F_{wt}]$  to the set of all those sets that share a non-empty intersection with the set constructed by  $[F_{wt}]$ .

In the Introduction we argued that a forged banknote is an (intended) approximation to a banknote. We also made the (simplistic) assumption that *being green* and *being made of polymer* exhaust *being a banknote*. Thus, one reason why  $a$  may be a forged banknote is because  $a$ , though being made of polymer, fails to be green. Therefore, at some  $\langle w, t \rangle$ ,  $a$  may have some, though not all, of the properties making up the essence of being a banknote,  $q$  ranging over  $(o\iota)_{\tau\omega}$ :

$$\lambda w \lambda t \ [ [ [Some \ \lambda q \ [q_{wt} \ a]] \ [essence \ banknote]] \wedge \neg [ [All \ \lambda q \ [q_{wt} \ a]] \ [essence \ banknote]] ] ]$$

A forged banknote is any individual that is not a banknote and which is either made of polymer but fails to be green, or is green but fails to be made of polymer. If we add a third property, e.g. *having a hologram*, it becomes an option that a non-banknote may have either one or two of those three properties and, therefore, qualify as a forged banknote to a lower or higher degree. Degrees of modification would be captured in TIL by spelling out which of the requisite properties of being a banknote a given forged banknote possessed.<sup>33</sup>

## References

1. Boldini, P.: Nonstandard type theory and natural language semantics. Presented at The Third International Tbilisi Symposium on Language, Logic and Computation, Batumi, Georgia (1999)
2. Duží, M., Jespersen, B., Materna, P.: Procedural Semantics for Hyperintensional Logic - Foundations and Applications of Transparent Intensional Logic. Series Logic, Epistemology and the Unity of Science. Springer, Berlin (forthcoming)
3. Jackendoff, R.: Semantic Structures. MIT Press, London, Cambridge (1990)
4. Jespersen, B.: Significant sententialism in Transparent Intensional Logic and Martin-Löf's Type Theory. In: T. Childers, O. Majer (eds.), The Logica Yearbook 2002, 117-131 (2003)
5. Jespersen, B.: Explicit intensionalization, anti-actualism, and how Smith's murderer might not have murdered Smith. *Dialectica* 59, 285-314 (2005)

<sup>32</sup> See [2] §1.4.3.

<sup>33</sup> Bjørn Jespersen is indebted to Marie Duží for very helpful suggestions regarding this subsection.

6. Jespersen, B.: Predication and extensionalization. *Journal of Philosophical Logic* 37, 479-99 (2008)
7. Jespersen, B.: Property modification and the rule of pseudo-detachment. (Manuscript)
8. Jespersen, B., Carrara, M.: Intensional logic for technical malfunction. (In submission)
9. Jespersen, B.: How hyper are hyperpropositions?. *Language and Linguistics Compass*. (Forthcoming)
10. Jespersen, B., Duží, M., Materna, P.: The *logos* of semantic structure. In: P. Stalmaszczyk (ed.), *Philosophy of Language and Linguistics. Volume I: The Formal Turn*. Ontos-Verlag, Farnkfurt. (Forthcoming)
11. Martin-Löf, P.: Constructive mathematics and computer programming. In: J.J. Cohen and others (eds.) *Sixth International Congress for Logic, methodology and Philosophy of Science*, 153-175. North-Holland, Amsterdam (1982)
12. Martin-Löf, P.: *Intuitionistic Type Theory*. Bibliopolis, Naples (1984).
13. Martin-Löf, P.: Mathematics of infinity. In: P. Martin-Löf and G. Mints (eds.) *Colog 88, International Conference on Computer Logic. Lecture Notes in Computer Science* 417, 146-197. Springer Verlag (1990)
14. Nordström, B., Petersson, K. and Smith, J.: *Programming in Martin-Löf's Type Theory*. Oxford University Press, Oxford (1990)
15. Primiero, G.: The determination of reference in a constructive setting. *Giornale di Metafisica*, 26, 3, 483-502 (2004)
16. Primiero, G.: Information and Knowledge - A constructive type-theoretical approach. *Series Logic Epistemology and the Unity of Science*, 10. Springer, Berlin (2008)
17. Primiero, G.: Proceeding in abstraction. From concepts to types and the recent perspective on information. *History and Philosophy of Logic*, 30, 257-282 (2009)
18. Ranta, A.: *Type Theoretical Grammar*. Oxford University Press, Oxford (1990)
19. Salvesen, A., Smith, J.: The strength of the subset type in Martin-Löf's Type Theory. In: *Proceedings of LICS'88*. IEEE, Edinburgh (1988)
20. Tichý, P.: *The Foundations of Frege's Logic*. De Gruyter, Berlin (1988)
21. Tichý, P.: *Collected Papers in Logic and Philosophy*. V. Svoboda, B. Jespersen and C. Cheyne (eds.). *Filosofia*, Czech Academy of Sciences, Prague; University of Otago Press, Dunedin (2004)